Deep networks
"Where did the variance go?"

David A. Johannsen David J. Marchette

Naval Surface Warfare Center – Dahlgren Division

17 May 2018

Acknowledgement: This work funded in part by the NSWC Naval Innovative Science and Engineering (NISE) program.

Motivation I

- "In practice, poor local minima are rarely a problem with large networks. Regardless of the initial conditions, the system nearly always reaches solutions of similar quality."

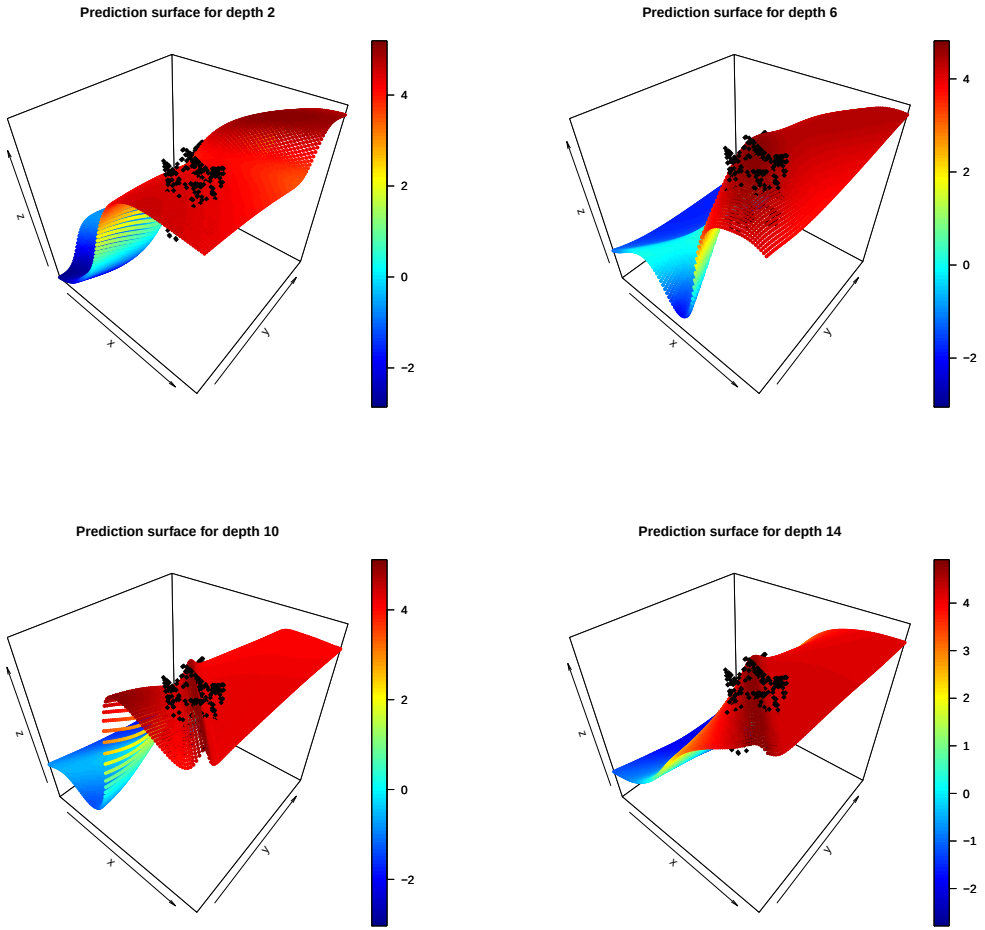
  — LeCun, Bengio, Hinton, "Deep Learning," *Nature*, 2015

- "..., while local minima are numerous, they are relatively easy to find, and they are all more or less equivalent in terms of performance on the test set."

  — Choromanska, Henaff, Mathieu, Ben Arous, LeCun, "The Loss Surfaces of Multilayer Networks," *Proc.* 18^{th} *Int. Conf. AISTATS, 2015*

Deep neural networks have many local minima and are "critical point indifferent:"
— What are the costs of (ever more) complicated models?

Motivation II



The polynomial anology and the Bias/Variance Trade-off:
— Where did the model complexity and variance go?

Our recurring example

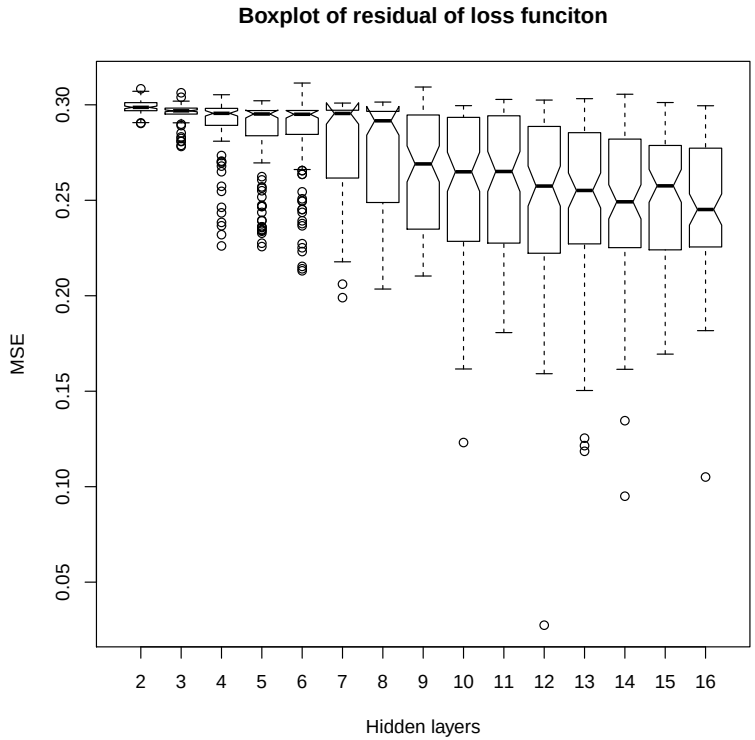
A description of the preceding example, as it will recur:

- "Vanilla" 2 — 8 — ... — 8 — 1 feed-forward neural network with tanh activation function.
- Experiments range from 2 to 16 hidden layers.
- Trained by back propagation to learn the function $f(x, y) = x + \cos(y^2) + \exp(\sin(x))$.
- Training data consists of 900 points from several uniformly sampled "blobs" in $-4 < x, y < 4$.
- Prediction surface is a grid in $-5 < x, y < 5$.

More of a regression/function approximation perspective for ease of illustration.

Our recurring example

Value of the loss function, with networks of varying depth (100 retrainings per depth).



- Residual is still decreasing.
- At 16 hidden layers, number of parameters (1113) has already eclipsed the cardinality of training data (900 points).

Where did the variance go?

Back to the motivating question:

Where did the variance go?

The answer:

Complicated network architectures may not yield a complicated model.

- Early convergence to flat saddle point means that the model may be closer to linear than a less complicated model — possibly yielding a more regular model.
- Much "residual randomness" left in the model — artifact of random initialization and stochastic elements of training.
- Away from training data, model predictions may be "almost" linear random projection.

Random model

So, complicated architectures may yield a simple, but random model.

- The simple part is appealing — Hand, *Classifier technology and the illusion of progress*. If you don't know much about the distribution or process(es) that gave rise to the data, then a simple model that regresses well is to be preferred.
- There is enough model variability to regress the training data, but no more.

The next several slides will present some evidence for the assertion that complicated networks may not produce a complicated model.

Back to the beginning — initialization

We will be considering typical initialization of "vanilla" feed-forward neural networks trained by back propogation.

- i) Inputs are scaled and centered.
- ii) Uniform adaptive initialization of the weight matrices, $U[-1/\sqrt{n_j}, 1/\sqrt{n_j}]$, or normalized initialization (Glorot and Bengio), $U[-\sqrt{6}/\sqrt{n_j + n_{j+1}}, \sqrt{6}/\sqrt{n_j + n_{j+1}}]$.
- Together i) and ii) ensure (with high probability) that inputs aren't out on the tails of the activation function.

Note that such initialization methods are essentially forced - gradients of tanh (or other sigmoid functions) quickly go to zero if one gets too far out on the tails.

Back to the beginning — initialization

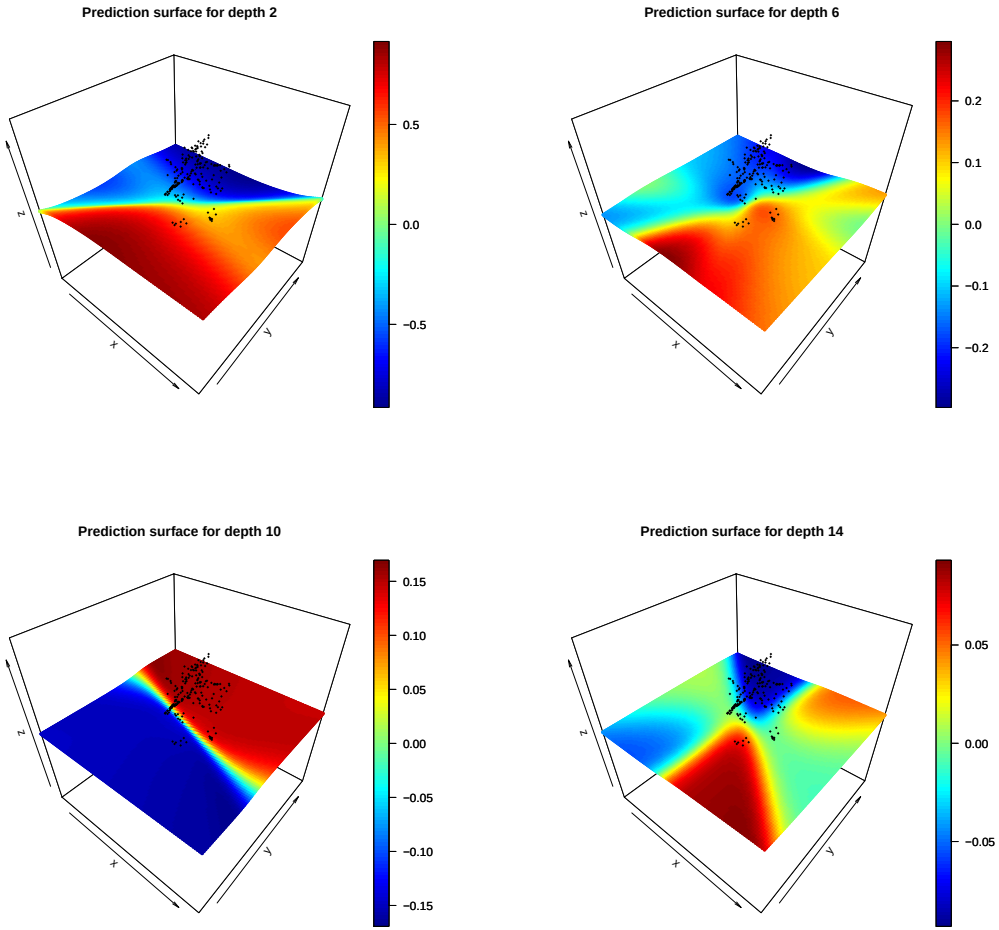
Continuing, we note that,

- iii) tanh (sigmoid functions, more generally) is an extremely good approximation to the identity function ($f(x) = x$) near the origin.

Together, i), ii), and iii) imply that, at initialization, a random linear map is not a bad approximation to a neural network.

Back to the beginning — initialization

At initialization, the network is "almost a linear map" (which becomes "more constant with growing depth").



The landscape (a.k.a., our assumptions)

In the remainder we accept the mounting evidence (Choromanska, Dauphin, Kawaguchi, Sankar, LeCun, etc.) that the following assertions are "generally (or often) true":

- the number of saddle points grows "combinatorially" with number of parameters;
- the landscape of typical loss functions is "flat" in the vicinity of saddles;
- stochastic gradient descent doesn't find local minima, but settles near one of the many saddle points (or at least a "flat minimum");
- these saddle points are all "almost" a global minimum for the loss function.

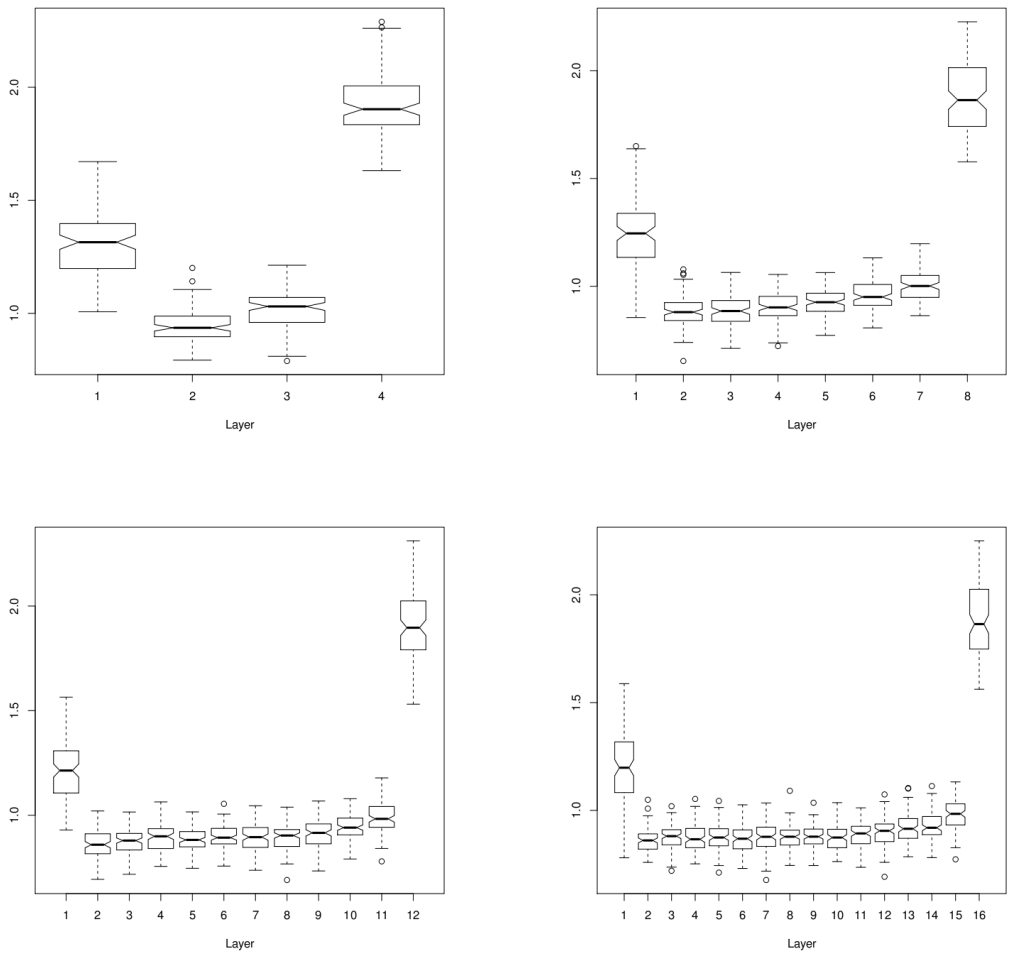
Bottom line: For whatever reason (one aspect of the "theory-practice gap"), converges to a point in feature space with small residual of the loss function.

Small weights

- If the weights are "small" (as they are at initialization) and one uses tanh (or other sigmoid) activation function, the map is not all that far from linear.
- Vanishing of gradient problem:
  - Entries in weight matrices (hence eigenvalues) are "small."
  - Chain rule for derivative mean that components of the gradient in early layers are a product of large number of small terms.
- In deep (or otherwise complicated) networks, the growing number of saddle points means that weights are likely to remain small at convergence, especially at the "front end" of a neural network.
- In fact, experiments indicate that "most of the work is done at the back end."
- So, can one replace a large portion of a deep neural network with a random linear projection (Johnson-Lindenstrauss)?

Small weights

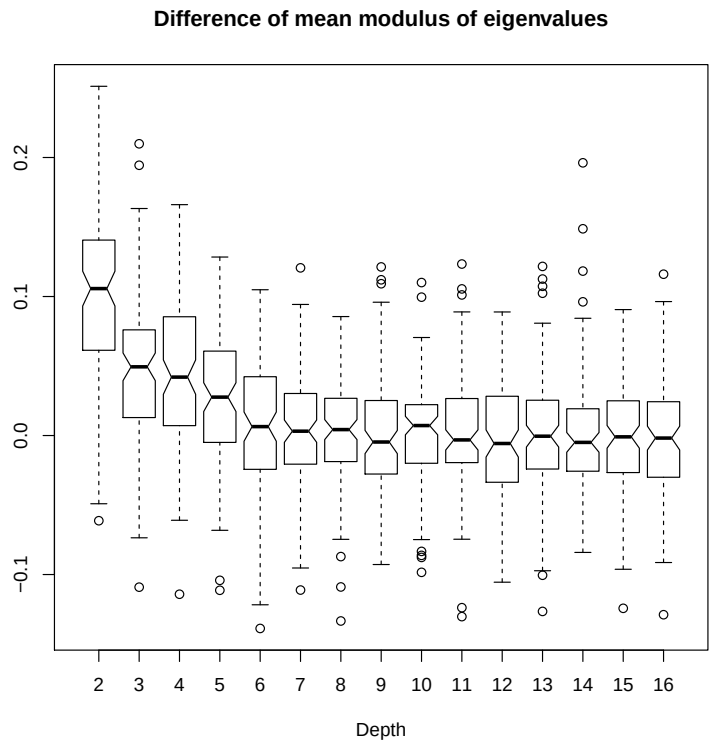
With growing depth, the mean absolute value of the singular values of the weight matrices at convergence decreases (especially "at the front end"):



Consequences of depth

The deeper the network, the smaller the difference between initialization and final weights.

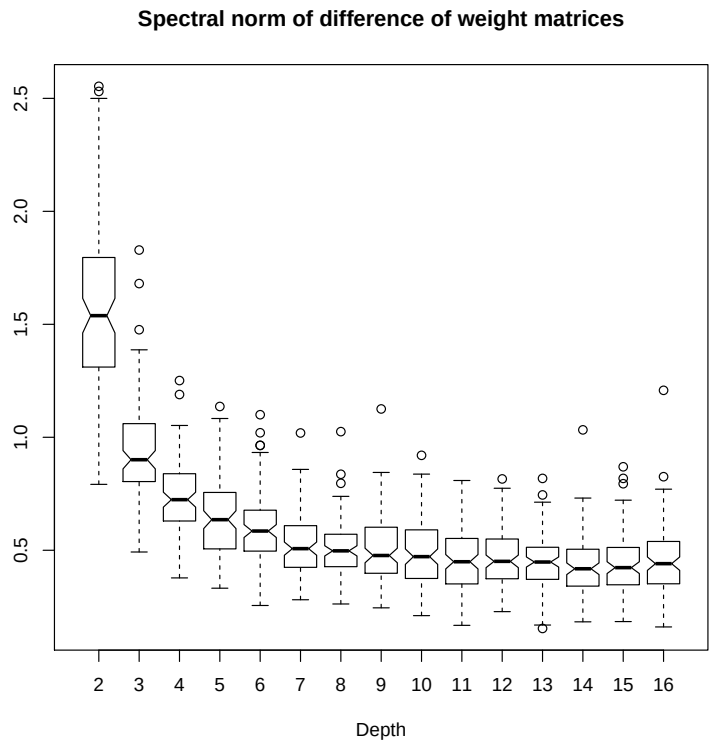
- Difference of mean modulus of eigenvalues between initial weight matrix and weights at convergence for the weights between first two hidden layers (100 retrainings):



Consequences of depth

The deeper the network, the smaller the difference between initialization and final weights.

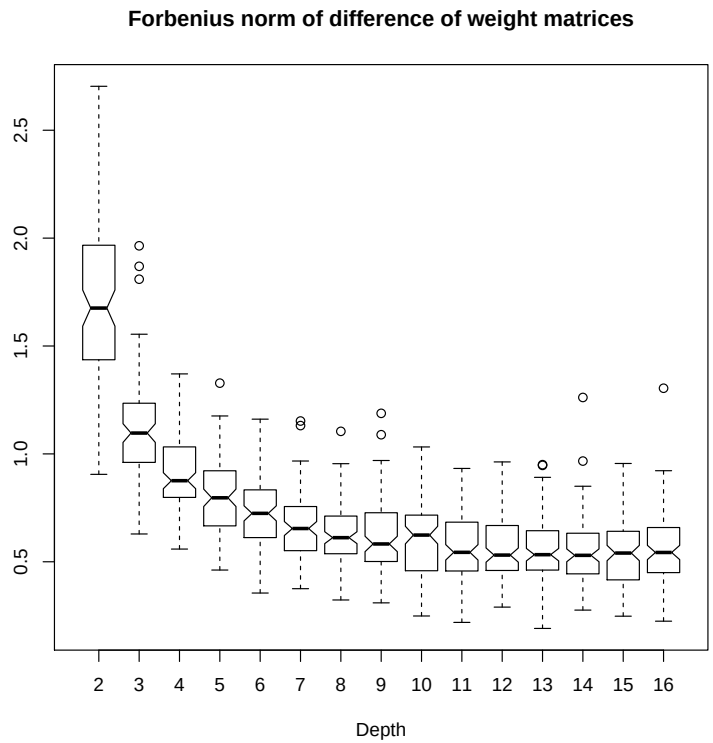
- Spectral norm of the difference between initial weight matrix and weights at convergence for the weights between the first two hidden layers (100 retrainings):



Consequences of depth

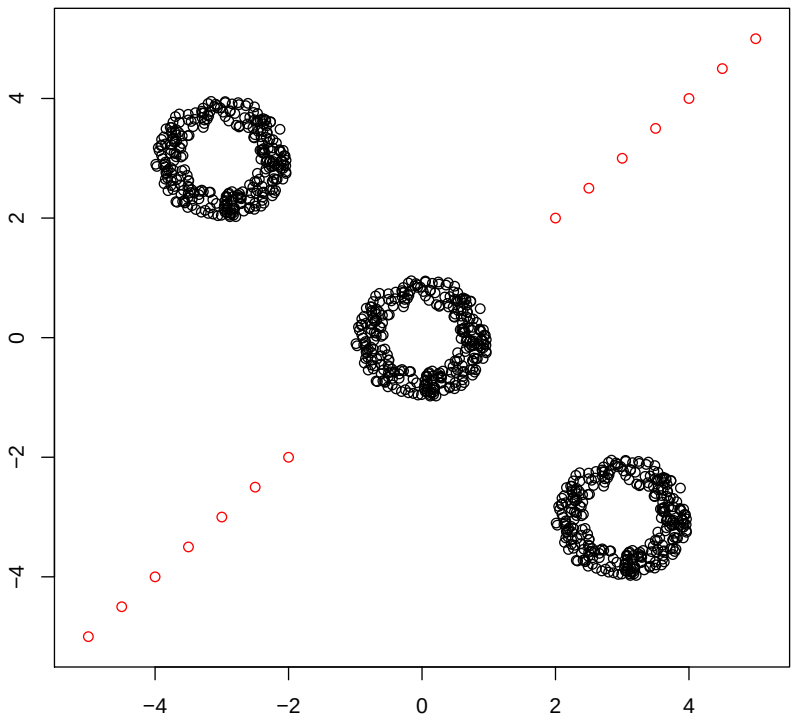
The deeper the network, the smaller the difference between initialization and final weights.

- Frobenius norm of the difference between initial weight matrix and weights at convergence for the weights between first two hidden layers (100 retrainings):



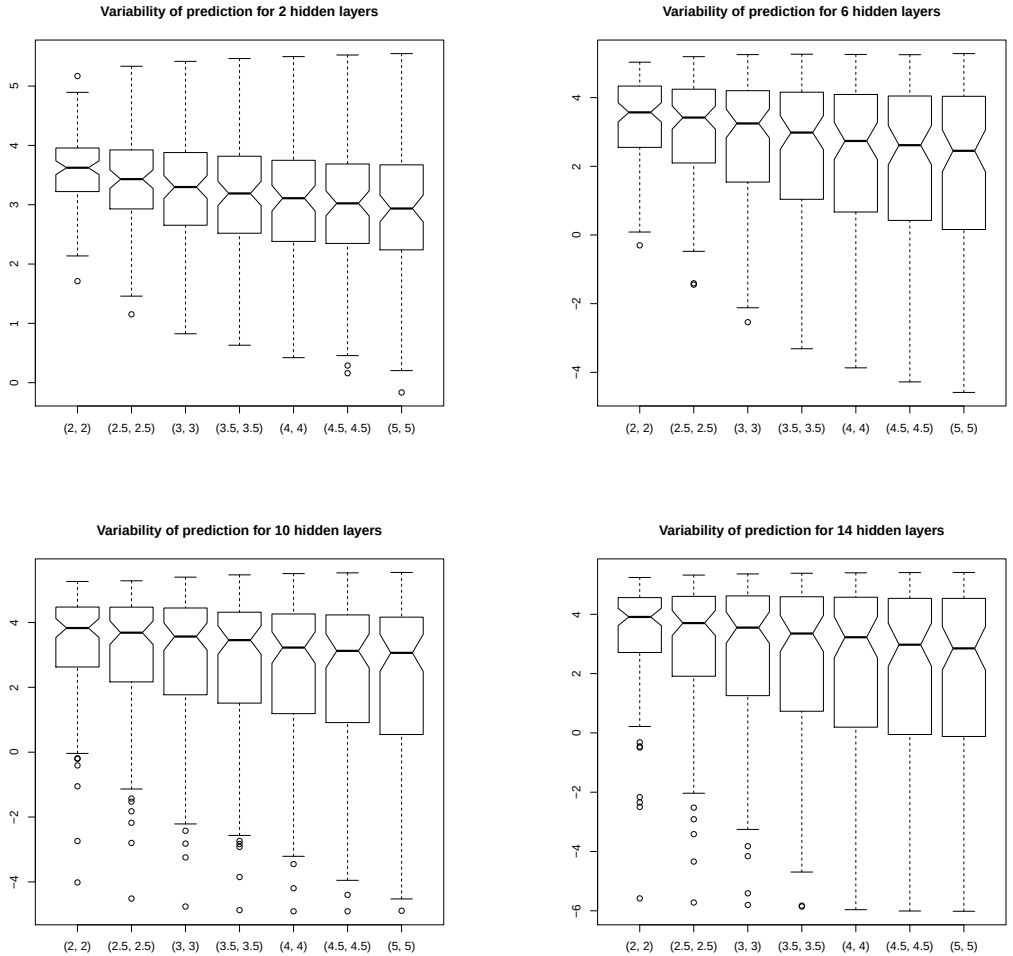
Model variability/variance

- As the distance of a test point grows from the training data, do complex models exhibit more prediction variability?
- Examined variability of predictions at each of the seven (in each quadrant) red points.



Retraining variability (1st quadrant)

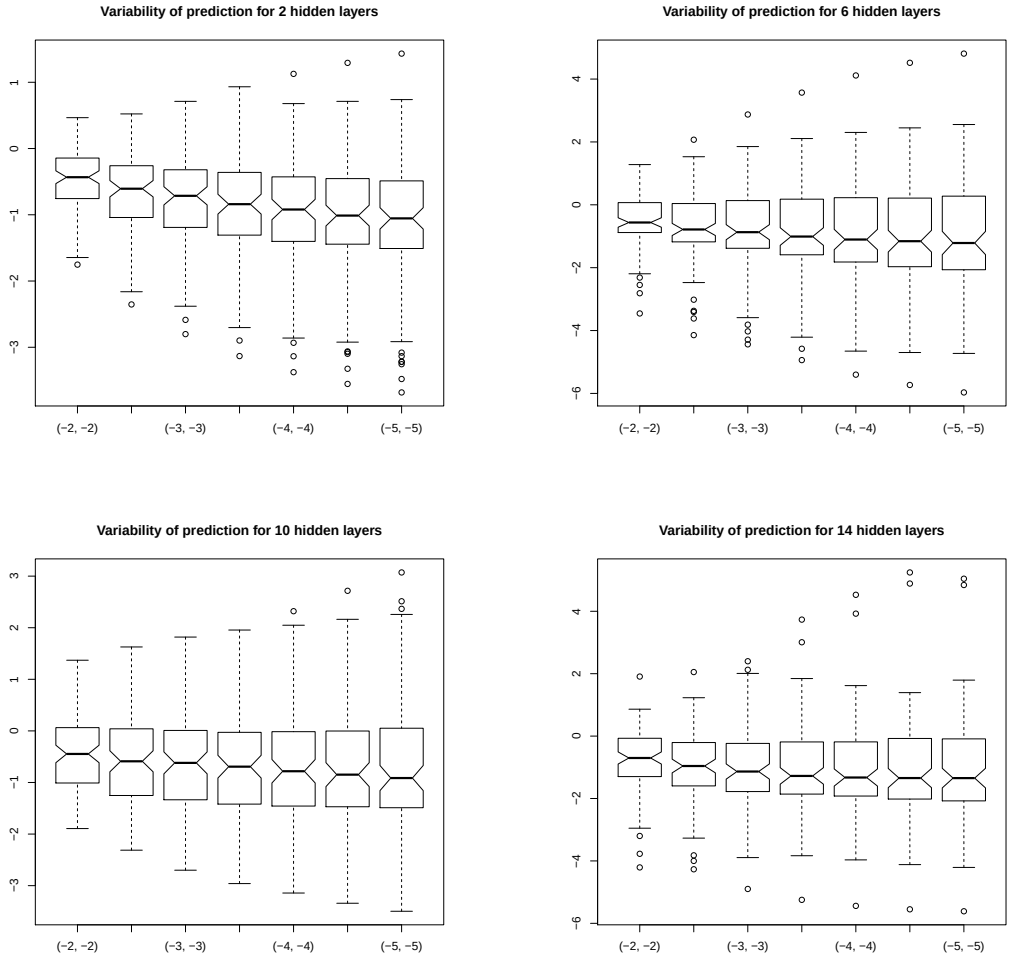
Box plots of variability of predictions over 100 retrainings.



Deep models do not exhibit significantly larger interquartile ranges.

Retraining variability (3rd quadrant)

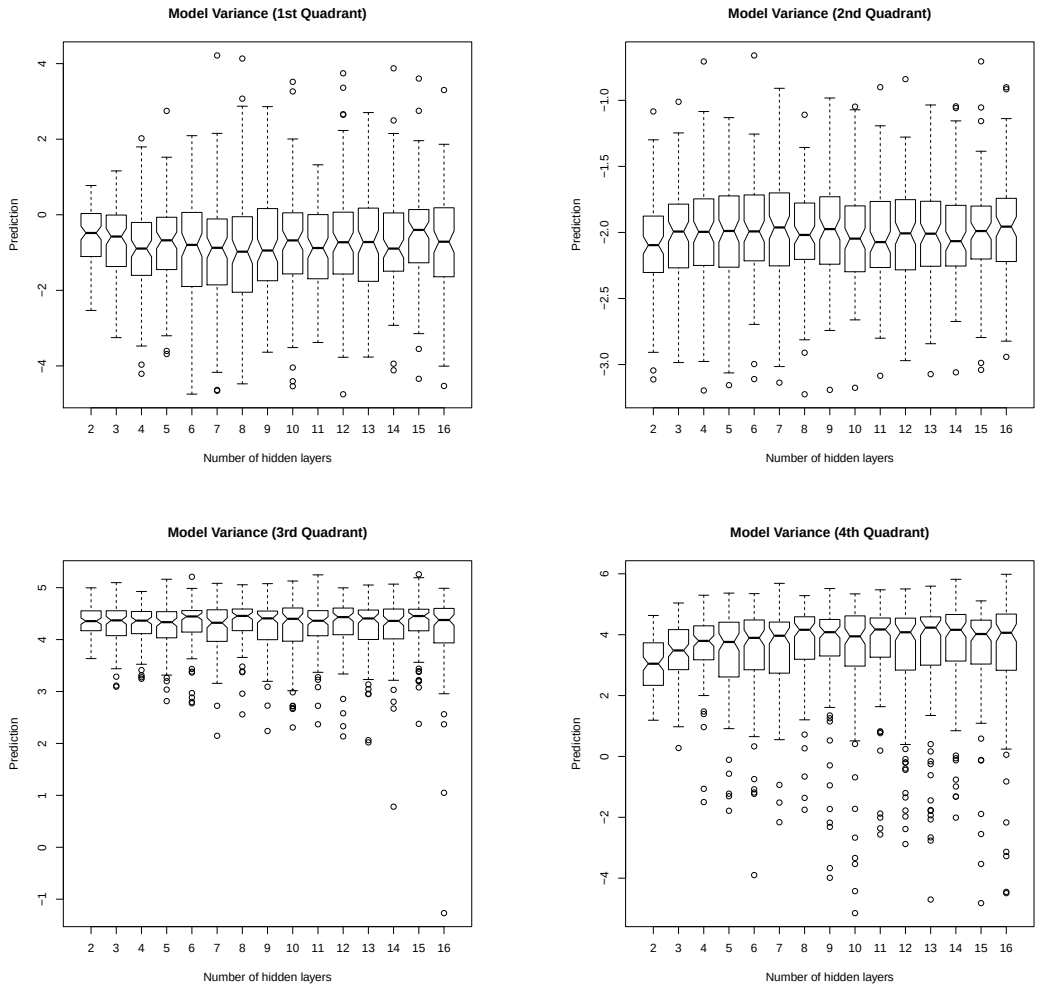
Box plots of variability of predictions over 100 retrainings.



Again, deep models do not exhibit larger interquartile ranges.

Model variance

Box plots of variability of predictions at $(\pm 5, \pm 5)$ over 100 draws of the training set (with subsequent training).



Deep models do not exhibit greater model variance.

Model variability/variance

- Examined two aspects of model variability: prediction variability with retraining, and model variance with new training data sampled from same distribution.
- Neither aspect of variability/variance increased significantly with model complexity.
- The more complex architecture does not yield a more complex model.

Conclusion

Conclusion: Complicated network architectures may not produce a complicated model.

- More complicated models tend to quickly settle into "flat minima."
- The quick convergence yields a simple model, many layers of which are not badly approximated by a random linear projection.
- Thus, more complicated models may exhibit more "regularity" than simple models and tend to generalize well.
- Over-fitting concerns might be exaggerated?
- The difference between initial and final weights has potential for a usable definition of "excess model capacity."